

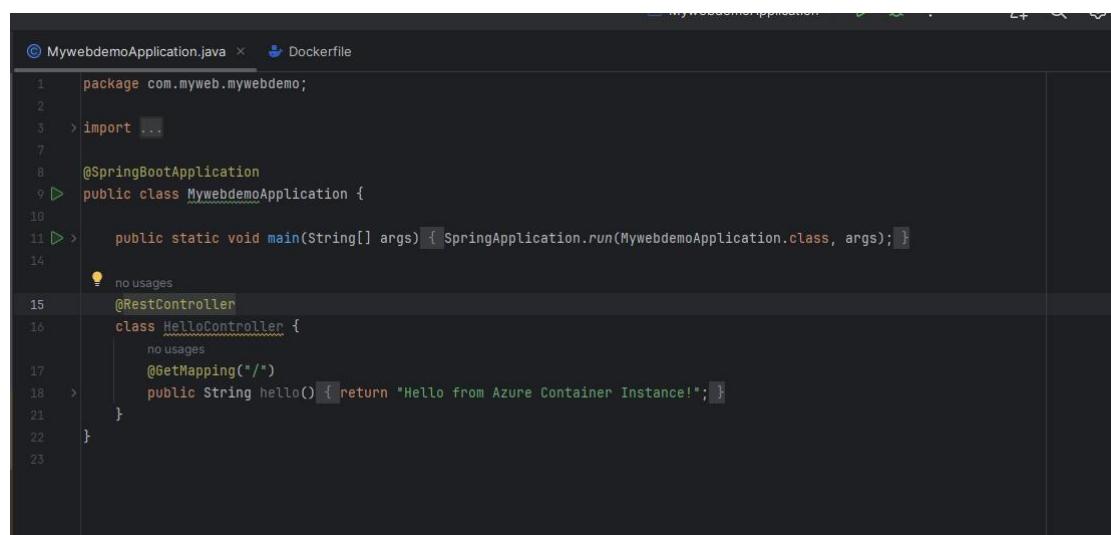
Practical Task 1: Deploy a Docker Container to Azure Container Instances (ACI) via Azure Portal

Requirements:

1. Create a lightweight Docker image for a simple web application (e.g., a Python Flask app) with minimal dependencies to reduce resource usage.
2. Push the Docker image to Azure Container Registry (ACR) using a low-cost storage option.
3. Use a lightweight ACI instance (e.g., B1s) to deploy the Docker container from ACR.
4. Verify the deployment by accessing the web application via the public IP address provided by ACI.
5. Remove the ACI container after verifying the deployment to stop billing.

Actions Taken:

1. Created a Docker image for a Java web application and pushed by docker



The screenshot shows a code editor with two tabs: 'MywebdemoApplication.java' and 'Dockerfile'. The Java file contains a main class 'MywebdemoApplication' with a main method and a controller class 'HelloController' with a single endpoint returning a hello message. The Dockerfile is a standard multi-stage build Dockerfile.

```
MywebdemoApplication.java
1 package com.myweb.mywebdemo;
2
3 > import ...
4
5 @SpringBootApplication
6 public class MywebdemoApplication {
7
8 >     public static void main(String[] args) { SpringApplication.run(MywebdemoApplication.class, args); }
9
10
11 >     @RestController
12     class HelloController {
13         no usages
14         @GetMapping("/")
15         public String hello() { return "Hello from Azure Container Instance!"; }
16     }
17
18 > }
19
20
21
22
23

Dockerfile
1 FROM maven:3.6.3-slim AS build
2 WORKDIR /app
3 COPY . .
4 RUN mvn clean package -DskipTests
5
6 FROM openjdk:8-jdk-alpine
7 ENV JAVA_OPTS=-Xms512M -Xmx1024M
8 COPY --from=build /app/target/*.jar app.jar
9 EXPOSE 8080
10 ENTRYPOINT ["java", "-Djava.security.egd=file:/dev/./urandom", "-jar", "app.jar"]
```

The screenshot shows a code editor with two tabs: 'MywebdemoApplication.java' and 'Dockerfile'. The 'Dockerfile' tab is active, displaying the following content:

```
1 FROM openjdk:17-jdk-slim
2 WORKDIR /app
3 COPY target/mywebdemo-0.0.1-SNAPSHOT.jar myapp.jar
4 EXPOSE 8080
5 CMD ["java", "-jar", "myapp.jar"]
```

2.Created ACR with name myjavaapp

The screenshot shows the 'Overview' page for the 'myjavaapp' container registry. Key details include:

- Resource group: (move)
- Location: West Europe
- Subscription: (move) Azure subscription 1
- Subscription ID: 9a6ae428-d8c3-44fe-bdf2-4e08593901a0
- Soft delete (Preview): Disabled
- Login server: myjavaapp.azurecr.io
- Creation date: 1/24/2025, 10:01 PM GMT+2
- Provisioning state: Succeeded
- Pricing plan: Basic

3.Created ACI with name myjavaapp

The screenshot shows the 'Overview' page for the 'myjavaapp' container instance. Key details include:

- Resource group: (move) : Volodymyr-Dibrova
- Status: Running
- Location: West Europe
- Subscription: (move) : Azure subscription 1
- Subscription ID: 9a6ae428-d8c3-44fe-bdf2-4e08593901a0
- Tags: (edit) : Add tags
- CPU: 100
- Memory: 1008

4.Tested in browser by public IP address



Practical Task 2: Configure Environment Variables in ACI via Azure Portal

Requirements:

1. Modify your Docker image to read configuration values from environment variables, ensuring minimal environmental complexity.
2. Reuse the ACI instance from Task 1 to deploy the container and specify the necessary environment variables.
3. Verify that the application is correctly using the environment variables by checking its output.
4. Remove the ACI container after verifying that the application correctly uses the environment variables.

Actions Taken:

1. Modified the Docker image to work with environment variables and pushed by docker

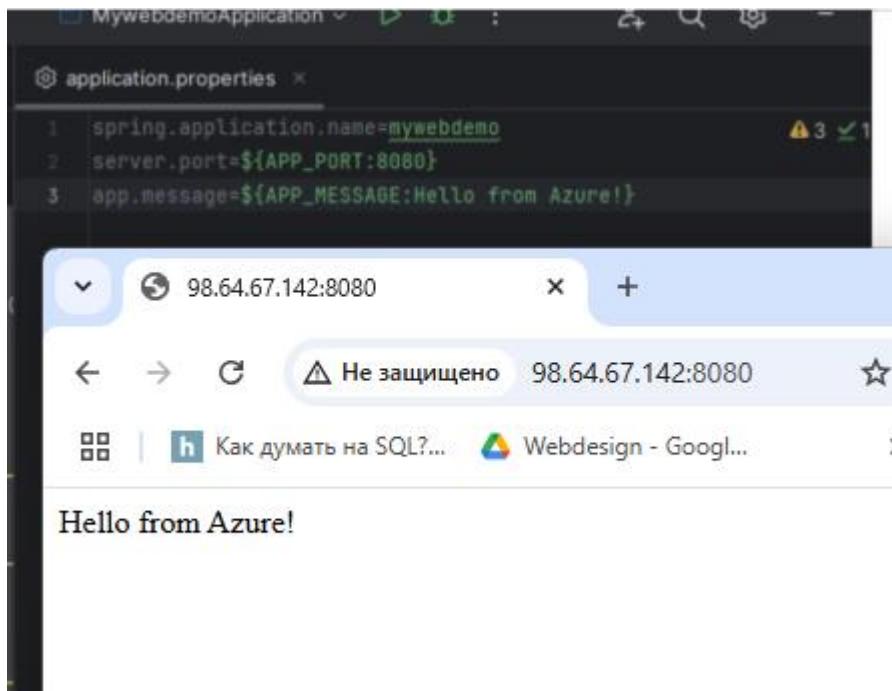
The screenshot shows an IDE with two files open:

- Mywebdemoadminstration.java**: A Spring Boot application class with a main method and a HelloController containing a hello() method that returns a String message.
- application.properties**: Configuration properties file with three entries: spring.application.name=mywebdemo, server.port=\${APP_PORT:8080}, and app.message=\${APP_MESSAGE>Hello from Azure!}.

```

PS C:\Users\User> docker tag mywebdemo myjavaapp.azurecr.io/mywebdemo:v2
PS C:\Users\User> docker push myjavaapp.azurecr.io/mywebdemo:v2
The push refers to repository [myjavaapp.azurecr.io/mywebdemo]
a547e969d43c: Waiting
0798cc789215: Waiting
7b3579d4413b: Waiting
1fe172e4850f: Waiting
44d3aa8d0766: Waiting
6ce99fdf16e8: Waiting
failed to authorize: failed to fetch oauth token: unexpected status from GET request to http://myjavaapp.azurecr.io/oauth2/token?scope=repository%3Amywebdemo%3Apull&scope(repository%3Amywebdemo%3Apull%2Cpush&service=myjavaapp.azurecr.io: 401 Unauthorized
PS C:\Users\User> az acr login --name myjavaapp
Login Succeeded
PS C:\Users\User> docker push myjavaapp.azurecr.io/mywebdemo:v2
The push refers to repository [myjavaapp.azurecr.io/mywebdemo]
44d3aa8d0766: Layer already exists
a547e969d43c: Layer already exists
0798cc789215: Pushed
7b3579d4413b: Pushed
6ce99fdf16e8: Layer already exists
1fe172e4850f: Layer already exists
v2: digest: sha256:43a5b6ac858fb8a242b2348d02c978693b621ee1f416796b3a95b7e3a335e7b6 size: 8
56
PS C:\Users\User> A

```



Practical Task 3: Scale Out with Azure Container Instances via Azure Portal

Requirements:

1. Deploy a stateless Docker container to Azure Container Instances using a lightweight configuration (e.g., B1s instances).
2. Manually scale out to the minimum number of instances required (e.g., 2–3) to test load distribution.
3. Stop all ACI instances after completing the testing to reduce ongoing costs.

Actions Taken:

First option: Without ACI

1. Created 3 containers in ACR

```

PS C:\Users\User> for ($x=1; $x -le 3; $x++) {
>>     az container create --resource-group Volodymyr-Dibrova `
>>         --name myjavaapp$x `
>>         --image myjavaapp.azurecr.io/mywebdemo:v1 `
>>         --cpu 1 --memory 1.5 --ports 8080 --os-type Linux `
>>         --registry-login-server myjavaapp.azurecr.io `
>>         --registry-username myjavaapp `
>>         --registry-password "rOxmOlIxoiuhLjq5v08a6Hr4hN90VuZL0i7+R69I6j+ACRA+Emo+"
>> }
{
    "confidentialComputeProperties": null,
    "containerGroupProfile": null,
    "containers": [
        {
            "command": null,
            "configMap": {
                "keyValuePairs": {}
            },
            "environmentVariables": [],
            "image": "myjavaapp.azurecr.io/mywebdemo:v1",
            "instanceView": {
                "currentState": {
                    "detailStatus": "",
                    "exitCode": null
                }
            }
        }
    ]
}
```

2.Checked logs for 3 containers

```

PS C:\Users\User> az container list --resource-group Volodymyr-Dibrova --output table
Name          ResourceGroup    Status     Image           IP:ports   Network  CPU/Memory  OsType  Location
myjavaapp1    Volodymyr-Dibrova Succeeded myjavapp.azuredcr.io/mywebdemo:v1  98.64.53.76:80,8080  Public   1.0 core/1.5 gb Linux   westeurope
myjavaapp1    Volodymyr-Dibrova Succeeded myjavapp.azuredcr.io/mywebdemo:v1  1.0 core/1.5 gb Linux   westeurope
myjavaapp2    Volodymyr-Dibrova Succeeded myjavapp.azuredcr.io/mywebdemo:v1  1.0 core/1.5 gb Linux   westeurope
myjavaapp3    Volodymyr-Dibrova Succeeded myjavapp.azuredcr.io/mywebdemo:v1  1.0 core/1.5 gb Linux   westeurope
PS C:\users\User> az container logs --resource-group Volodymyr-Dibrova --name myjavaapp1

:: Spring Boot ::      (v3.3.8-SNAPSHOT)

2025-01-25T10:40:27.479Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT using Java 17
2025-01-25T10:40:27.491Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 default profile: "dev"
2025-01-25T10:40:31.834Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s) 8080 (http)
2025-01-25T10:40:31.906Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-25T10:40:31.987Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.34]
2025-01-25T10:40:32.427Z INFO 19 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext

```

```
PS C:\Users\User> az container logs --resource-group Volodymyr-Dirovva --name myjavaapp2

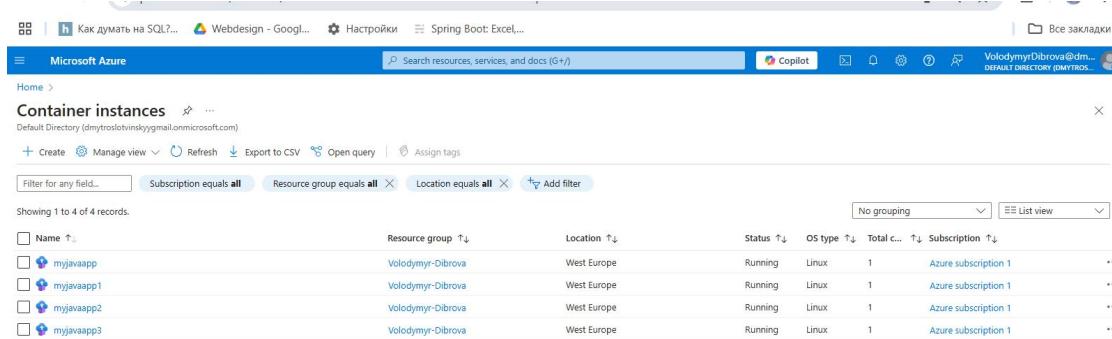
:: Spring Boot ::          (v3.3.8-SNAPSHOT)

2025-01-25T10:49:29.842Z  INFO 19 — [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 19 (/app/myapp.jar starte
d by root in /app)
2025-01-25T10:49:29.850Z  INFO 19 — [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Starting Tomcat Web Server
2025-01-25T10:49:32.699Z  INFO 19 — [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8088 (HTTP)
2025-01-25T10:49:32.758Z  INFO 19 — [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-25T10:49:32.759Z  INFO 19 — [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.34]
2025-01-25T10:49:32.760Z  INFO 19 — [mywebdemo] [main] o.a.c.t.Catalina@localhost:8088 : [Tomcat]: initialized in [0] ms
2025-01-25T10:49:33.221Z  INFO 19 — [mywebdemo] [main] w.s.s.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3944 ms
2025-01-25T10:49:35.135Z  INFO 19 — [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8088 (http) with context path '/'
2025-01-25T10:49:35.231Z  INFO 19 — [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 7.91 seconds (process running for 10.108)
```

```
PS C:\Users\Users\az container logs --resource-group Volodymyr-Dibrova --name myjavaapp3

:: Spring Boot ::   (v3.3.8-SNAPSHOT)

2025-01-25T10:50:38,149Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 19 (/app/myapp.jar started by root in /app)
2025-01-25T10:50:38,189Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 default profile: "default"
2025-01-25T10:50:42,567Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8088 (http)
2025-01-25T10:50:42,651Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-25T10:50:42,653Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.0.20]
2025-01-25T10:50:42,654Z INFO 19 --- [mywebdemo] [main] o.a.catalina.startup.ContextHandler : Initializing [ROOT] [localhost:8088]
2025-01-25T10:50:43,138Z INFO 19 --- [mywebdemo] [main] o.s.w.s.DevToolsApplicationContext : Root WebApplicationContext: initialization completed in 4787 ms
2025-01-25T10:50:45,297Z INFO 19 --- [mywebdemo] [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8088 (http) with context path '/'
2025-01-25T10:50:45,487Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 9.593 seconds (process running for 12.987)
PS C:\Users\Users\|
```



3. Tested in browser



Second option: with ACI

```
1 provider "azurerm" {
2     features {}
3     subscription_id = "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
4 }
5
6 data "azurerm_resource_group" "rg" {
7     name = "Volodymyr-Dibrova"
8 }
9
10 resource "azurerm_kubernetes_cluster" "aks" {
11     name          = var.aks_cluster_name
12     location      = "westeurope"
13     resource_group_name = data.azurerm_resource_group.rg.name
14     dns_prefix    = "myaksdns"
15
16     default_node_pool {
17         name      = "default"
18         node_count = 3
19         vm_size   = "Standard_D2_v2"
20     }
21
22     identity {
23         type = "SystemAssigned"
24     }
25
26     tags = {
27         environment = "dev"
28     }
29 }
30
31 output "kube_config" {
32     value    = azurerm_kubernetes_cluster.aks.kube_config_raw
33     sensitive = true
34 }
35
```

```
PS E:\projects\terraform\Proj\terraform-aks> az aks get-credentials --resource-group Volodymyr-Dirova --name myakscluster --overwrite-existing
Merged "myakscluster" as current context in C:\Users\USER\.kube\config
PS E:\projects\terraform\Proj\terraform-aks> kubectl get nodes
NAME           STATUS    ROLES   AGE     VERSION
aks-default-88566151-vms000000  Ready    <none>  9m37s  v1.30.7
aks-default-88566151-vms000001  Ready    <none>  9m40s  v1.30.7
aks-default-88566151-vms000002  Ready    <none>  9m37s  v1.30.7
PS E:\projects\terraform\Proj\terraform-aks> |
```

The screenshot shows the Azure Kubernetes Services blade for the 'myakscluster' cluster. The left sidebar lists cluster-level navigation options like Overview, Activity log, Access control (IAM), Tags, Monitor, Diagnose and solve problems, Microsoft Defender for Cloud (preview), Cost analysis, and Kubernetes resources (Namespaces, Workloads). The main area displays the 'Services and ingresses' blade, which includes a search bar, a 'Create' button, and a refresh button. It shows a table of services with columns for Name, Namespace, Status, Type, Cluster IP, External IP, Ports, and Age. The table contains entries for 'kubernetes' (Namespace: default, Status: Ok, Type: ClusterIP, IP: 10.0.0.1, Ports: 443/TCP, Age: 14 minutes), 'kube-dns' (Namespace: kube-system, Status: Ok, Type: ClusterIP, IP: 10.0.0.10, Ports: 53/UDP, 53/TCP, Age: 13 minutes), and 'metrics-server' (Namespace: kube-system, Status: Ok, Type: ClusterIP, IP: 10.0.141.72, Ports: 443/TCP, Age: 13 minutes).

Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP	14 minutes
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP, 53/TCP	13 minutes
metrics-server	kube-system	Ok	ClusterIP	10.0.141.72		443/TCP	13 minutes

```

PS E:\projects\terraformProj> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster --overwrite-existing
Merged "myakscluster" as current context in C:\Users\User\.kube\config
PS E:\projects\terraformProj> kubectl get nodes
NAME STATUS ROLES AGE VERSION
aks-default-88566151~vmss000000 Ready <none> 9m37s v1.30.7
aks-default-88566151~vmss000001 Ready <none> 9m40s v1.30.7
aks-default-88566151~vmss000002 Ready <none> 9m37s v1.30.7
PS E:\projects\terraformProj> az acr credential show --name myjavaapp
{
  "passwords": [
    {
      "name": "password",
      "value": "rOxm0lIxoiuhLjq5v08a6Hr4hN90VuZLoi7+R69I6j+ACRA+Emo+"
    },
    {
      "name": "password2",
      "value": "aJLardyzCEPBBKwdH197wOxbfu/jVTEqc3xMmSBta+ACRBZwPKB"
    }
  ],
  "username": "myjavaapp"
}
PS E:\projects\terraformProj> kubectl create secret docker-registry acr-secret \
--docker-server=myjavaapp.azurecr.io \
--docker-username=myjavaapp \
--docker-password="rOxm0lIxoiuhLjq5v08a6Hr4hN90VuZLoi7+R69I6j+ACRA+Emo+"
secret/acr-secret created
PS E:\projects\terraformProj> |
```

PS E:\projects\terraformProj\terraform> kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
mywebdemo-deployment-dc9b99db4-5t2dv	0/1	ContainerCreating	0	5s
mywebdemo-deployment-dc9b99db4-cls9d	0/1	ContainerCreating	0	5s
mywebdemo-deployment-dc9b99db4-qhc fz	0/1	ContainerCreating	0	5s

PS E:\projects\terraformProj\terraform> kubectl get deployments

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
mywebdemo-deployment	3/3	3	3	13s

Logs for every containers:

```

PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-5t2dv
[:: Spring Boot ::] (v3.3.8-SNAPSHOT)
2025-01-29T14:50:39.442Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 /app/myapp.jar started by root in /app
2025-01-29T14:50:39.455Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.792Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.809Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.810Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.021Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.025Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1476 ms
2025-01-29T14:50:42.135Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.163Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.409 seconds
(process running for 4.178)
PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-cls9d
[:: Spring Boot ::] (v3.3.8-SNAPSHOT)
```

```

PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-cls9d
[:: Spring Boot ::] (v3.3.8-SNAPSHOT)
2025-01-29T14:50:39.366Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 /app/myapp.jar started by root in /app
2025-01-29T14:50:39.370Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.849Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.869Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.869Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.043Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.045Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1604 ms
2025-01-29T14:50:42.072Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.090Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.256 seconds
(process running for 4.065)
2025-01-29T14:50:47.408Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet 'dispatc
herServlet'
2025-01-29T14:50:47.409Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-01-29T14:50:47.410Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-qhc fz
[:: Spring Boot ::] (v3.3.8-SNAPSHOT)
```

```

PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-qhcfcz

:: Spring Boot ::      (v3.3.8-SNAPSHOT)
2025-01-29T14:50:39.614Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 /app/myapp.jar started by root in /app
2025-01-29T14:50:39.617Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.945Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.962Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.963Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.242Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.244Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1568 ms
2025-01-29T14:50:42.174Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.194Z INFO 0 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.195 seconds
(process running for 3.922)
2025-01-29T14:59:58.660Z INFO 0 --- [mywebdemo] [nio-8080-exec-4] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet 'dispatc
herServlet'
2025-01-29T14:59:58.660Z INFO 0 --- [mywebdemo] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-01-29T14:59:58.662Z INFO 0 --- [mywebdemo] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
PS E:\projects\terraformProj\nodeWebApp>

```

Check load balancing between 3 pods

```

PS E:\projects\terraformProj\nodeWebApp> while ($true) { curl http://40.74.47.105; Start-Sleep -Seconds 1 }

StatusDescription : StatusCode : 200
RawContent      : Content      : Hello from Azure!
                  StatusDescription :
Forms           : Content      : HTTP/1.1 200
Headers         : Content      : Keep-Alive: timeout=60
                  Content      : Connection: keep-alive
                  Content      : Content-Length: 17
                  Content      : Content-Type: text/plain;charset=UTF-8
                  Content      : Date: Wed, 29 Jan 2025 15:08:06 GMT
                  Content      : Hello from Azure!
                  StatusDescription :
                  Content      : HTTP/1.1 200
                  Content      : Keep-Alive, timeout=60
                  Content      : Connection, keep-alive
                  Content      : Content-Length, 17
                  Content      : Content-Type, text/plain;charset=UTF-8...
                  Content      : Date, Wed, 29 Jan 2025 15:08:06 GMT
                  Content      : Hello from Azure!
Forms           : Content      : []
Headers         : Content      : []
Images          : Content      : []
InputFields     : Content      : []
Links           : Content      : []
Object          : Content      : []
RawContentLength : ParsedHtml : System.__Com
StatusDescription : StatusCode : 200
RawContent      : Content      : Hello from Azure!
                  StatusDescription :
Forms           : Content      : HTTP/1.1 200
Headers         : Content      : Content-Length: 17
                  Content      : Content-Type: text/plain;charset=UTF-8
                  Content      : Date: Wed, 29 Jan 2025 15:08:07 GMT
                  Content      : Hello from Azure!
                  Content      : []
                  StatusDescription :
                  Content      : HTTP/1.1 200
                  Content      : Content-Length: 17
                  Content      : Content-Type: text/plain;charset=UTF-8
                  Content      : Date: Wed, 29 Jan 2025 15:08:07 GMT
                  Content      : Hello from Azure!
Forms           : Content      : []
Headers         : Content      : []
Images          : Content      : []
InputFields     : Content      : []
Links           : Content      : []
Object          : Content      : []
RawContentLength : ParsedHtml : System.__Com
StatusDescription : StatusCode : 200
RawContent      : Content      : Hello From Azure!
                  StatusDescription :
Forms           : Content      : HTTP/1.1 200
Headers         : Content      : Content-Length: 17
                  Content      : Content-Type: text/plain;charset=UTF-8
                  Content      : Date: Wed, 29 Jan 2025 15:08:09 GMT
                  Content      : Hello From Azure!
                  Content      : []
                  StatusDescription :
                  Content      : HTTP/1.1 200
                  Content      : Content-Length: 17
                  Content      : Content-Type: text/plain;charset=UTF-8
                  Content      : Date: Wed, 29 Jan 2025 15:08:09 GMT
                  Content      : Hello From Azure!
Forms           : Content      : []
Headers         : Content      : []
Images          : Content      : []
InputFields     : Content      : []
Links           : Content      : []
Object          : Content      : []
RawContentLength : ParsedHtml : System.__Com

```

Practical Task 4: Secure a Docker Container in ACI with Managed Identity via Azure Portal

Requirements:

- Deploy a Docker container to Azure Container Instances using the existing lightweight ACI setup from previous tasks.
- Configure a Managed Identity for the ACI and securely access an Azure service (e.g., Azure Key Vault) with minimal permissions and access scope.
- Retrieve only a single secret from Azure Key Vault for testing purposes.
- Remove the ACI container after verifying secure access.

Actions Taken:

1.Created app for getting Secret from Azure

The screenshot shows a code editor with two tabs: `MywebdemoApplication.java` and `application.properties`.

```

public class SecretController {
    private final SecretClient secretClient;

    public SecretController(SecretClient secretClient) {
        this.secretClient = secretClient;
    }

    @GetMapping("/secret")
    public String getSecret() {
        try {
            return secretClient.getSecret("name: MySecret").getValue();
        } catch (Exception e) {
            return "Failed to retrieve secret.";
        }
    }
}

```

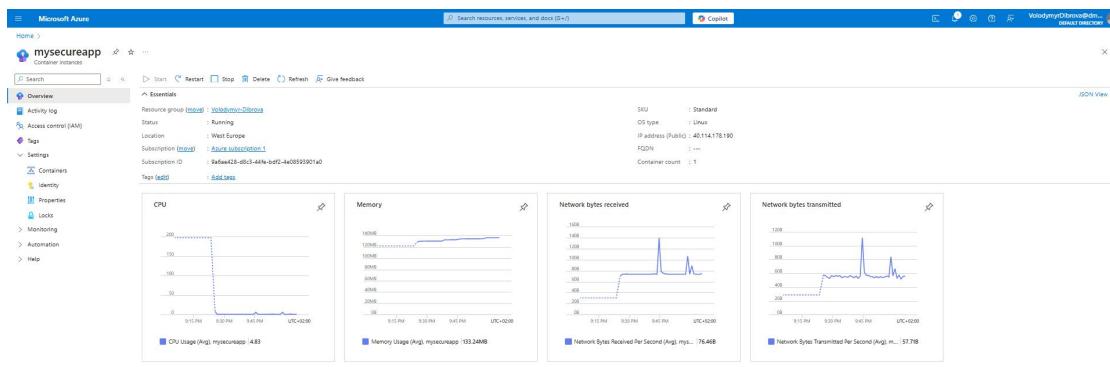
`application.properties` content:

```

spring.application.name=mywebdemo
server.port=${APP_PORT:8080}
app.message=${APP_MESSAGE:Hello from Azure!}

azure.keyvault.uri=https://mykeyvault2025.vault.azure.net
azure.keyvault.secret=MySecret

```



2.Created Key Vault and added Role Assignment (Key Vault Administrator)

The screenshot shows the Azure portal's Key Vault blade for the `myKeyVault2025` key vault.

Essentials section:

- Resource group: `mywebdemo-2025`
- Location: West Europe
- Subscription: `my subscription`
- Subscription ID: `90e4a220-0b13-44fe-bd21-4a02059301a0`
- Tags: `add tags`

Properties section:

- Vault URL: `https://mykeyvault2025.vault.azure.net/`
- State (Holding key): Standard
- Directory ID: `8f1157ba-19f4-415f-8240-a0fa29485d78`
- Directory Name: Default Directory
- Soft-delete: Enabled
- Purge protection: Disabled

Manage keys and secrets used by apps and services section:

- Control access to key vault: Assign access policy and determine which application principal, namely an application or user group, can perform different operations on key vault keys.
- Enable logging and set up alerts: Enable logging to monitor how often keys are used and turn on alerts for performance and configuration alerts.
- Turn on recovery options: For database agents, enable soft-deletes. If enabled, turn on purge protection to guard against manual deletion of deleted keys.

The screenshot shows the Azure portal's Access control (IAM) blade for the `myKeyVault2025` key vault.

Action	Principal	Role	Subscription	Notes	
Add	Search	Download role assignments	Edit columns	Refresh	
Add	Dmytro Slovinskyj (dmytro.slovinskyj@gmail.com#EXT#@)	User	Owner	Subscription (Inherited)	None
Add	Foreign Principal for FOREIGN CO	Foreign principal	Owner	Subscription (Inherited)	None
Add	Capgemini Azure Course 2025 (ca)	Group	Contributor	Subscription (Inherited)	None
Add	besaha-principal1 (besaha-principal1)	App	Advisor Reviews Reader	Subscription (Inherited)	None
Add	besaha-principal (besaha-principal)	App	Compute Recommendations Role	Subscription (Inherited)	None
Add	Volodymyr Dibrova (VolodymyrDibrova@dmytroslo...)	User	Key Vault Administrator	This resource	None

Properties

- Created: 1/25/2022, 9:33:08 PM
- Updated: 1/25/2022, 9:33:08 PM
- Secret identifier: <https://mykeyvault20221125.cust.azure.net/secrets/708baa7b46ad431fbda96f32bfb1bc7>
- Settings:
 - Set activation date:
 - Set expiration date:
- Enabled: Yes
- Tags: 0 tags
- Secret Content type (optional):
- Secret value: SuperSecretValue

Tags: 0 tags

Secret

Content type (optional)

Hide Secret Value

Secret value: SuperSecretValue

deployed v3 in mywebdemo

myjavaapp | Repositories

mywebdemo

Tag	Digest	Last modified
v1	sha256:98f2eae0c1400020909c53aee16ec0d8fd1a295045a5546c33009974d0f125	1/25/2022, 10:03 PM GMT+2
v2	sha256:d1a108ea5586ea22b212d02c9709210214e161797602a907a7a1215a7b8	1/25/2022, 12:18 PM GMT+2
v3	sha256:bd7709c1e544de301a485da732283984d2909ea34ca7e9f7a7d1111121ca	1/24/2022, 10:18 PM GMT+2

Tested in browser by v3 - retrieved secret

108.141.3.117:8080/secret

SuperSecretValue

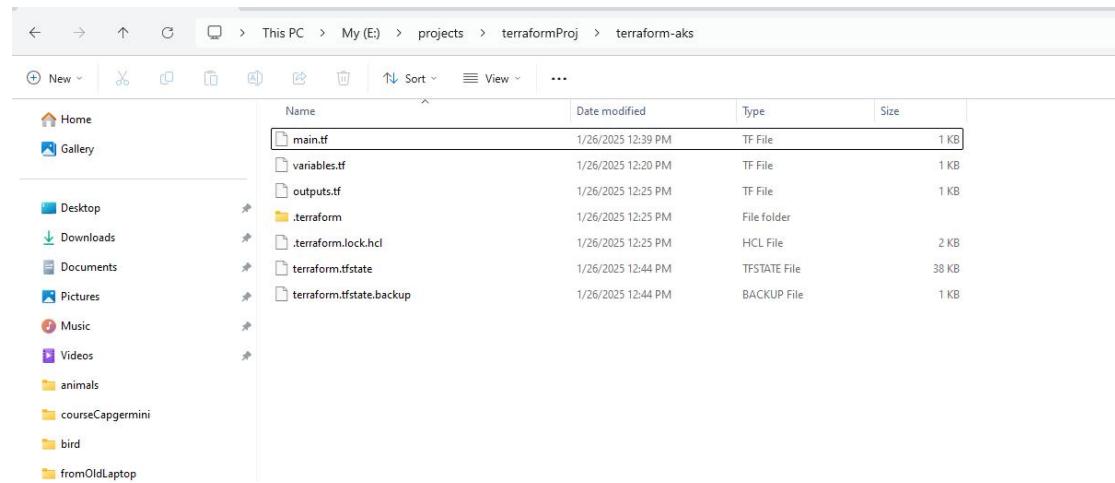
Practical Task 5: Deploy a Kubernetes Cluster with AKS via Azure Portal Requirements:

1. Create an Azure Kubernetes Service (AKS) cluster with the smallest VM size (e.g., B2s) and the minimum number of nodes (e.g., 1–2).
2. Connect to the AKS cluster using Azure Cloud Shell with kubectl.

3. Deploy a lightweight Nginx application for verification.
4. Delete the AKS cluster immediately after testing to avoid additional VM and cluster

Actions Taken:

- 1.created cluster using Terraform



The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The left sidebar contains icons for file operations like Open, Save, Find, and others. The main workspace has tabs for Welcome, main.tf (which is the active tab), variables.tf, and outputs.tf. The code editor displays the following Terraform configuration:

```
E: > projects > terraformProj > terraform-aks > main.tf
1 provider "azurerm" {
2   features {}
3   subscription_id = "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
4 }
5
6 data "azurerm_resource_group" "rg" {
7   name = "Volodymyr-Dibrova"
8 }
9
10 resource "azurerm_kubernetes_cluster" "aks" [
11   name          = var.aks_cluster_name
12   location      = "westeurope"
13   resource_group_name = "Volodymyr-Dibrova"
14   dns_prefix    = "myaksdns"
15 ]
16
17   default_node_pool {
18     name        = "default"
19     node_count = var.node_count
20     vm_size    = var.vm_size
21   }
22
23   identity {
24     type = "SystemAssigned"
25   }
26
27   tags = {
28     environment = "dev"
29   }
30 ]
31
32
33 output "kube_config" {
34   value    = azurerm_kubernetes_cluster.aks.kube_config_raw
35   sensitive = true
36 }
37
```

File Edit Selection View Go Run Terminal Help

Welcome main.tf variables.tf ● outputs.tf

E: > projects > terraformProj > terraform-aks > variables.tf

```
1
2 variable "aks_cluster_name" {
3   default = "myakscluster"
4 }
5
6 variable "location" {
7   default = "West Europe"
8 }
9
10 variable "node_count" {
11   default = 1
12 }
13
14 variable "vm_size" {
15   default = "Standard_B2s"
16 }
17
```

File Edit Selection View Go Run Terminal Help

Welcome main.tf variables.tf ● outputs.tf X

E: > projects > terraformProj > terraform-aks > outputs.tf

```
1
2 output "aks_fqdn" {
3   value = azurerm_kubernetes_cluster.aks.fqdn
4 }
5
```

Microsoft Azure

Kubernetes services

myakscluster

Overview

Activity log

Tags

Diagnose and solve problems

Microsoft Defender for Cloud (preview)

Cost analysis

Kubernetes resources

Get started Properties Monitoring Capabilities (S) Recommendations (0) Tutorials

Networking

Setting	Value
API server address	myakscluster-hub1.https://westeurope.azurefdn.io
Network configuration	Auto-assign IP
Pod CIDR	10.244.0.0/16
Service CIDR	10.0.0.10
DNS service IP	10.0.0.10
Cilium dataplane	Not enabled
Network Policy	None

Successfully connected

```
PS /home/volodymyr> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster
Merged "myakscluster" as current context in /home/volodymyr/.kube/config
PS /home/volodymyr> kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
aks-default-26255206-vmss000000   Ready    <none>    13m   v1.30.7
PS /home/volodymyr>
```

The screenshot shows a code editor with several tabs at the top: Welcome, main.tf, variables.tf, nginx-deployment.yaml (which is the active tab), and nginx-service.yaml. The code in the editor is a Kubernetes Deployment manifest:

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: nginx-deployment
5   labels:
6     app: nginx
7 spec:
8   replicas: 2
9   selector:
10    matchLabels:
11      app: nginx
12   template:
13     metadata:
14       labels:
15         app: nginx
16     spec:
17       containers:
18         - name: nginx
19           image: nginx:latest
20           ports:
21             - containerPort: 80
```

The screenshot shows a code editor with several tabs at the top: Welcome, main.tf, variables.tf, nginx-deployment.yaml, and nginx-service.yaml (which is the active tab). The code in the editor is a Kubernetes Service manifest:

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: nginx-service
5 spec:
6   selector:
7     app: nginx
8   ports:
9     - protocol: TCP
10    port: 80
11    targetPort: 80
12   type: LoadBalancer
```

```

PS E:\projects\terraformProj\terraform-aks> kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
PS E:\projects\terraformProj\terraform-aks> kubectl apply -f nginx-service.yaml
service/nginx-service created
PS E:\projects\terraformProj\terraform-aks> kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           41s
PS E:\projects\terraformProj\terraform-aks> kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-deployment-576c6b7b6-6pkt6   1/1     Running   0          53s
nginx-deployment-576c6b7b6-zbqq4   1/1     Running   0          53s
PS E:\projects\terraformProj\terraform-aks> kubectl get services
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
kubernetes      ClusterIP   10.0.0.1      <none>         443/TCP       24m
nginx-service   LoadBalancer 10.0.184.157  20.8.228.57    80:31934/TCP  33s
PS E:\projects\terraformProj\terraform-aks> kubectl get svc nginx-service
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
nginx-service   LoadBalancer 10.0.184.157  20.8.228.57    80:31934/TCP  44s
PS E:\projects\terraformProj\terraform-aks> |

```

Nginx Deployment:

Name: nginx-deployment

Status: Both pods (2/2) are **running correctly**

Age: 41 seconds.

Pods (containers):

Both pods are running (Running)

Age: 53 seconds.

Nginx Service:

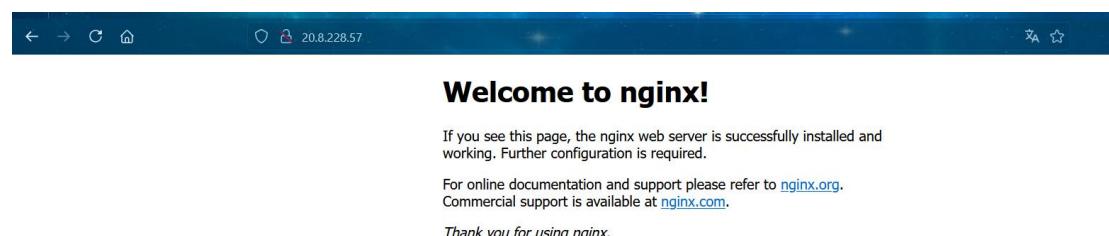
Type: LoadBalancer

Internal IP (CLUSTER-IP): 10.0.184.157

External IP (EXTERNAL-IP): **20.8.228.57**

Port: 80:31934/TCP

Age: 44 seconds.

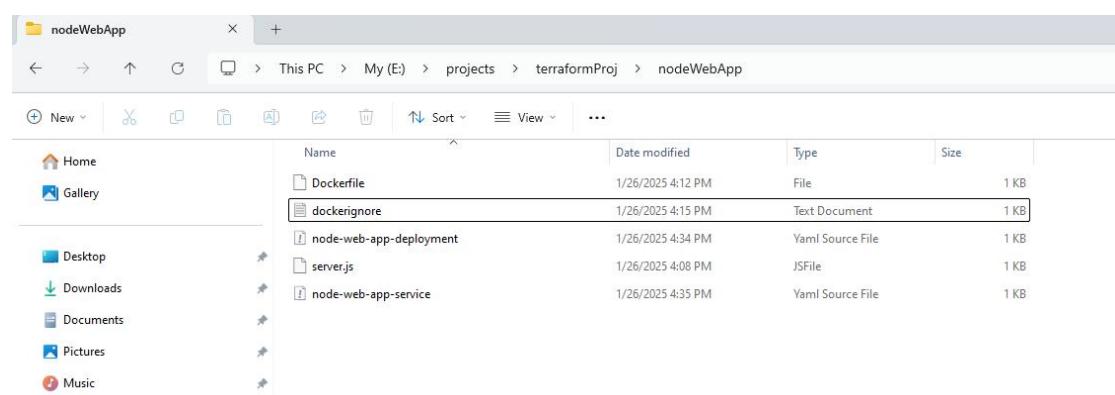


Practical Task 6: Deploy a Containerized Application on AKS

Requirements:

1. Build a lightweight Docker image for a simple web application (e.g., a Node.js app with minimal dependencies) and push it to Azure Container Registry (ACR).
2. Reuse the AKS cluster from Task 5 to deploy the application using a Kubernetes deployment and service manifest file.
3. Test the application for a limited time and remove the deployment afterward.

Actions Taken:



```
PS E:\projects\terraformProj\nodeWebApp> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
myjavaapp.azurecr.io/node-web-app   v1      43242ddf49e0  2 minutes ago  181MB
<none>            <none>    dfa7d3ec182d  18 hours ago  722MB
mywebdemo          latest    dd8b1164f06f  18 hours ago  722MB
myjavaapp.azurecr.io/mywebdemo     v3      dd8b1164f06f  18 hours ago  722MB
myjavaapp.azurecr.io/mywebdemo     v2      9f82baefe0c1  18 hours ago  722MB
<none>            <none>    43a5b6ac858f  28 hours ago  680MB
myapp              latest    b5f709c1e5e4  43 hours ago  680MB
myjavaapp.azurecr.io/mywebdemo     v1      b5f709c1e5e4  43 hours ago  680MB
workerapi-gateway      latest    59b2d00c1d86  5 weeks ago   681MB
worker             latest    2d050d945d91  5 weeks ago   681MB
frontend           latest    d7bee6c4b85e  5 weeks ago   1.36GB
gcr.io/k8s-minikube/kicbase      v0.0.45  e7c9bc3bc515  4 months ago  1.81GB
gcr.io/k8s-minikube/kicbase      <none>   81df28859520  4 months ago  1.81GB
PS E:\projects\terraformProj\nodeWebApp> docker push myjavaapp.azurecr.io/node-web-app:v1
The push refers to repository [myjavaapp.azurecr.io/node-web-app]
6504e29600c8: Waiting
c1875295cf1f: Waiting
5650d6de56fd: Waiting
d85d1128dd96: Waiting
4ad5ea901fd1: Waiting
37892ffbfcaa: Waiting
1f3e46996e29: Waiting
failed to authorize: failed to fetch oauth token: unexpected status from GET request to https://myjavaapp.azurecr.io/v1/_auth/login?service=mywebdemo
Error response from daemon: unauthorized
PS E:\projects\terraformProj\nodeWebApp> az acr login --name myjavaapp
Login Succeeded
PS E:\projects\terraformProj\nodeWebApp> docker push myjavaapp.azurecr.io/node-web-app:v1
The push refers to repository [myjavaapp.azurecr.io/node-web-app]
5650d6de56fd: Pushed
c1875295cf1f: Pushed
1f3e46996e29: Pushed
d85d1128dd96: Pushed
4ad5ea901fd1: Pushed
37892ffbfcaa: Pushed
6504e29600c8: Pushed
v1: digest: sha256:43242ddf49e080d7dd55b603fe355155885581e1a807c54a14b6bc252889e12d size: 856
```

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: node-web-app
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: node-web-app
10     template:
11       metadata:
12         labels:
13           app: node-web-app
14       spec:
15         containers:
16           - name: node-web-app
17             image: myjavaapp.azurecr.io/node-web-app:v1
18             ports:
19               - containerPort: 3000
20

```

Microsoft Azure

myjavaapp | Repositories

Home > myjavaapp > Repositories

Search resources, services, and docs (S+)

Copy

Activity log

Access control (IAM)

Tags

Quick start

Events

Settings

Services

Repositories

- Webhooks
- Geo-replications
- Tasks
- Connected registries
- (Previous)

Search

Refresh

Manage Deleted Repositories

Search to filter repositories ...

Repositories (1)

Cache Rule

mywebdemo

node-web-app

Microsoft Azure

Home > myk8scuster > Workloads > node-web-app | Overview

Deployment

Search

Refresh

View Grafana workspace

Give feedback

Configure

Troubleshoot issues with advanced infrastructure signals, alerts, and Grafana dashboards.

Overview

YAML

Events

Live logs

Essentials

Namespace: default

Ready: 2/2

Revision history: 10

Strategy type: RollingUpdate

Labels: app=

Selector: app=node-web-app

Cluster: myk8scuster

Creation time: 2021-01-27T16:04:23.000Z

Min ready seconds: 0

Replicas: 2 desired, 2 updated, 2 total, 2 available, 0 unavailable

Rolling update strategy: 25% max unavailable, 25% max surge

See more

Events

Total: 2

Warning: 0

Normal: 2

Workloads

Pods

Replica sets

Deleted

Show labels

Pod name	Ready	Status	Restart count	Age	CPU	Memory	Node	Pod IP
node-web-app-d87655db-4icp	1/1	Running	0	12 minutes	Enable metrics	0%	aks-default-227...	10.244.0.036
node-web-app-d87655db-7g7c	1/1	Running	0	12 minutes	0%	0%	aks-default-227...	10.244.0.117

Resource utilization

```
PS E:\projects\terraformProj\nodeWebApp> kubectl cluster-info
Kubernetes control plane is running at https://myaksdns-7txr3evi.hcp.westeurope.azurek8s.io:443
CoreDNS is running at https://myaksdns-7txr3evi.hcp.westeurope.azurek8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://myaksdns-7txr3evi.hcp.westeurope.azurek8s.io:443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS E:\projects\terraformProj\nodeWebApp> az acr repository list --name myjavaapp --output table
Result
mywebdemo
node-web-app
PS E:\projects\terraformProj\nodeWebApp> kubectl apply -f node-web-app-deployment.yaml
deployment.apps/node-web-app created
PS E:\projects\terraformProj\nodeWebApp> kubectl apply -f node-web-app-service.yaml
service/node-web-app-service created
PS E:\projects\terraformProj\nodeWebApp> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
node-web-app   0/2     2           0           18s
PS E:\projects\terraformProj\nodeWebApp> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
node-web-app-d876558db-z5nnk   0/1     ImagePullBackOff   0      25s
node-web-app-d876558db-zrdgg   0/1     ImagePullBackOff   0      25s
PS E:\projects\terraformProj\nodeWebApp> kubectl get services
NAME        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.0.0.1    <none>        443/TCP   11m
node-web-app-service   LoadBalancer  10.0.115.85  57.153.38.238  80:31347/TCP  25s
PS E:\projects\terraformProj\nodeWebApp> kubectl get svc node-web-app-service
NAME        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
node-web-app-service   LoadBalancer  10.0.115.85  57.153.38.238  80:31347/TCP  39s
PS E:\projects\terraformProj\nodeWebApp> |
```



Hello from Node.js app on AKS!

Practical Task 7: Configure and Use ConfigMaps and Secrets in AKS

Requirements:

1. Create a ConfigMap to store non-sensitive configuration data with only the required key value pairs for the application.
2. Create a Kubernetes Secret to store sensitive data (e.g., API keys) with the least amount of information needed.
3. Update the application deployment to use the ConfigMap and Secret.
4. Remove the ConfigMap, Secret, and deployment after testing.

Actions Taken:

```

apiVersion: v1
kind: Secret
metadata:
  name: my-app-secret
  namespace: default
  type: Opaque
data:
  DB_PASSWORD: dm12a==
  API_TOKEN: c2VjcmV0G0r7W4=
PS E:\projects\terraformProj\nodeWebApp> kubectl apply -f configmap.yaml
configmap/my-app-config created
PS E:\projects\terraformProj\nodeWebApp> kubectl get configmap my-app-config -o yaml
apiVersion: v1
data:
  API_URL: https://api.myapp.com
  APP_NAME: MyAKSApp
  LOG_LEVEL: info
kind: ConfigMap
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration:
      {"apiVersion":"v1","data":{"API_URL":"https://api.myapp.com","APP_NAME":"MyAKSApp","LOG_LEVEL":"info"},"kind":"ConfigMap"}
  creationTimestamp: 2025-01-26T19:16:22Z
  name: my-app-config
  namespace: default
  resourceVersion: "22389"
  uid: de7ccr04-93d7-41ba-91fe-8301102c7b05
dn12a==
PS E:\projects\terraformProj\nodeWebApp> [Convert]::ToString([Text.Encoding]::UTF8.GetBytes("vivi"))
PS E:\projects\terraformProj\nodeWebApp> [Convert]::ToString([Text.Encoding]::UTF8.GetBytes("secrettoken"))
c2VjcmV0G0r7W4=
PS E:\projects\terraformProj\nodeWebApp>

```

```
PS E:\projects\terraformProj\terraform-aks> kubectl create configmap app-config --from-literal=APP_NAME="Node Web Application" --from-literal=APP_ENV="production" --from-literal=LOG_LEVEL="info"
```

```
PS E:\projects\terraformProj\terraform-aks> kubectl create secret generic app-secret --from-literal=DB_PASSWORD="SuperSecretPassword" --from-literal=API_KEY="12345-ABCDE"
```

```

resource "azurerm_kubernetes_cluster" "aks" {
  name                = var.aks_cluster_name
  location            = "westeurope"
  resource_group_name = "Volodymyr-Dibrova"
  dns_prefix          = "myaksdns"

  default_node_pool {
    name        = "default"
    node_count = var.node_count
    vm_size    = var.vm_size
  }

  identity {
    type = "SystemAssigned"
  }

  tags = {
    environment = "dev"
  }
}

output "kube_config" {
  value     = azurerm_kubernetes_cluster.aks.kube_config_raw
  sensitive = true
}

resource "kubernetes_config_map" "app_config" {
  metadata {
    name = "app-config"
    namespace = "default"
  }

  data = {
    APP_NAME  = "Node Web Application"
    APP_ENV   = "production"
    LOG_LEVEL = "info"
  }
}

resource "kubernetes_secret" "app_secret" {
  metadata {
    name = "app-secret"
    namespace = "default"
  }

  data = {
    DB_PASSWORD = "SuperSecretPassword"
    API_KEY    = "12345-ABCDE"
  }

  type = "Opaque"
}

```

API_KEY	MItisDUQUIDREU=
DB_PASSWORD	U3VwZDUITZWNgZWRQYNNzsl9yZA==

Screenshot of a Microsoft Azure DevOps pipeline interface showing multiple tabs open (configmap.yaml, secret.yaml, deployment.yaml, main.tf, variables.tf, node-web-app-deployment.yaml) and a code editor displaying the deployment configuration.

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: node-web-app
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: node-web-app
10     template:
11       metadata:
12         labels:
13           app: node-web-app
14       spec:
15         containers:
16           - name: node-web-app
17             image: myjavaapp.azurecr.io/node-web-app:v1
18             ports:
19               - containerPort: 8080
20             env:
21               - name: APP_NAME
22                 valueFrom:
23                   configMapKeyRef:
24                     name: app-config
25                     key: APP_NAME
26               - name: DB_PASSWORD
27                 valueFrom:
28                   secretKeyRef:
29                     name: app-secret
30                     key: DB_PASSWORD
31

```

Below the code editor is the Microsoft Azure portal "node-web-app | Overview" page. It shows the deployment status (2 ready, 2 total), logs, events (2 total, 0 warnings, 2 normal), and pods (2 running). A terminal window at the bottom shows the command-line output of the deployment process.

```

Apply complete! Resources: 1 added, 1 changed, 0 destroyed.

Outputs:
aks_fqdn = "myaksdns-n72ja54b.hcp.westeurope.azmk8s.io"
kube_config = <sensitive>
PS E:\projects\terraformProj\terraform-aks> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
node-web-app   2/2     2           2           23s
PS E:\projects\terraformProj\terraform-aks> kubectl logs deployment/node-web-app
Found 2 pods, using pod/node-web-app-57596f7f98-vpwqd
Server is running on port 3000
PS E:\projects\terraformProj\terraform-aks> kubectl get services
NAME            TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes      ClusterIP      10.0.0.1    <none>        443/TCP   3h54m
PS E:\projects\terraformProj\terraform-aks> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
node-web-app-57596f7f98-vpwqd  1/1     Running   0          11m
node-web-app-57596f7f98-xvtjj  1/1     Running   0          11m
PS E:\projects\terraformProj\terraform-aks> kubectl exec -it node-web-app-57596f7f98-vpwqd -- printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=node-web-app-57596f7f98-vpwqd
NODE_VERSION=18.20.6
YARN_VERSION=1.22.22
APP_NAME=Node Web Application
DB_PASSWORD=SuperSecretPassword
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.0.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.0.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.0.0.1
KUBERNETES_SERVICE_HOST=10.0.0.1
KUBERNETES_SERVICE_PORT=443
TERM=xterm
HOME=/root
PS E:\projects\terraformProj\terraform-aks> |

```

The screenshot shows the Azure portal interface for a deployment named 'node-web-app'. Key details include:

- Deployment Overview:** Namespace: default, Revision history: 10, Strategy type: RollingUpdate, Labels: app=node-web-app.
- Events:** Total: 2, Warning: 0, Normal: 2.
- Pods:** Two pods are listed: node-web-app-37199799-rcpod and node-web-app-37199799-ing. Both are in a Running state with 1/1 readiness.
- Resource utilization:** CPU and Memory usage for both pods is shown, along with their respective Node and Pod IP addresses.

8.Practical Task 8: Scale Applications in AKS

Requirements:

1. Deploy a stateless application to the AKS cluster using minimal resource specifications.
2. Use the kubectl scale command to manually scale the application to only 2–3 replicas for testing.
3. Set up Horizontal Pod Autoscaler (HPA) with reasonable CPU usage thresholds to minimize pod creation.
4. Simulate load on the application for a short duration and remove the deployment after observing the scaling behavior

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: stateless-app
  labels:
    app: stateless-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: stateless-app
  template:
    metadata:
      labels:
        app: stateless-app
    spec:
      containers:
        - name: stateless-app
          image: nginx
          resources:
            requests:
              cpu: "250m"
              memory: "128Mi"
            limits:
              cpu: "500m"
              memory: "256Mi"
          ports:
            - containerPort: 80

```

The screenshot shows a code editor with several tabs open, including configmap.yaml, secret.yaml, deployment.yaml, main.tf, variables.tf, node-web-app-deployment.yaml, and stateless-app.yaml. The stateless-app.yaml tab is active, displaying the provided YAML configuration for a deployment named 'stateless-app' with a single replica. The configuration includes a selector matching the 'app: stateless-app' label, a template with a single container (nginx), and resource requests and limits for CPU and memory. A red box highlights the 'requests' section of the container configuration.

configmap.yaml.yaml

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: stateless-app
5  spec:
6    selector:
7      app: stateless-app
8    ports:
9      - protocol: TCP
10         port: 80
11         targetPort: 80
12

```

Microsoft Azure

stateless-app | Overview

Deployment

Overview

YAML

Events

Live logs

Essentials

Namespace: default

Ready: 1/1

Revision history: 10

Strategy type: RollingUpdate

Labels: app=stateless-app

Selector: app=stateless-app

Cluster: myakscluster

Creation time: 2025-01-27T18:43:17.000Z

Min ready seconds: 0

Replicas: 1 desired, 1 updated, 1 total, 1 available, 0 unavailable

Rolling update strategy: 25% max unavailable, 25% max surge

Events

Total: 0

Warning: 0

Normal: 0

Resource utilization

Workloads

Pods

Pod name	Ready	Status	Restart count	Age	CPU	Memory	Node	Pod IP
stateless-app-79c79dd5-4274t	1/1	Running	0	13 hours	8%	1%	aks-default-108...	10.244.0.62

```

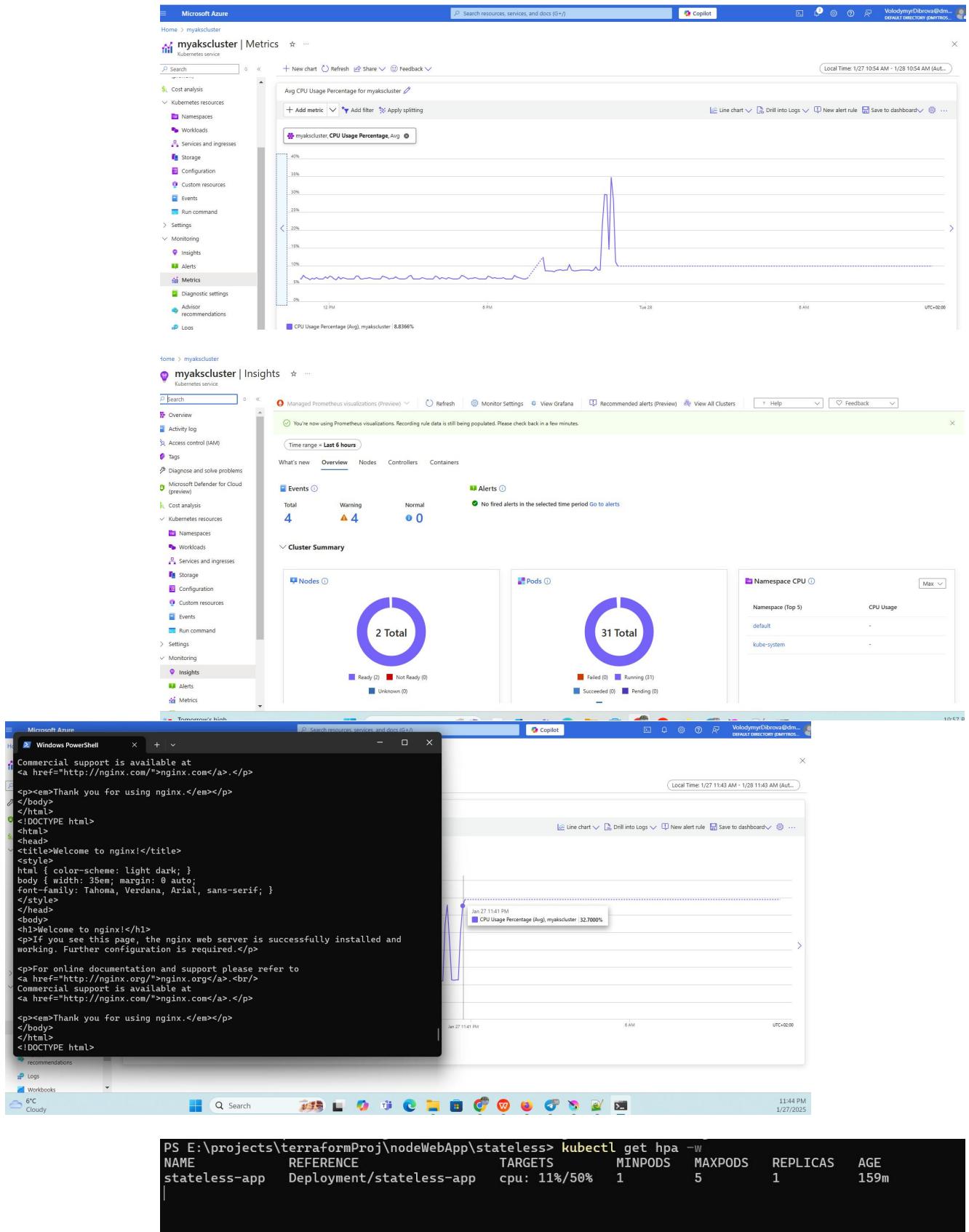
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl apply -f stateless-app.yaml
deployment.apps/stateless-app created
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
stateless-app 1/1     1           1           17m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get deployment stateless-app
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
stateless-app 1/1     1           1           19m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl scale deployment stateless-app --replicas=3
deployment.apps/stateless-app scaled
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get deployment stateless-app
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
stateless-app 3/3     3           3           20m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get pods -l app=stateless-app
NAME                READY   STATUS    RESTARTS   AGE
stateless-app-79c79dd5-5chgz  1/1     Running   0          31s
stateless-app-79c79dd5-c6g2j  1/1     Running   0          31s
stateless-app-79c79dd5-hl67q  1/1     Running   0          20m
PS E:\projects\terraformProj\nodeWebApp\stateless>

```

```

PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get pods -l app=stateless-app
NAME                READY   STATUS    RESTARTS   AGE
stateless-app-79c79dd5-5chgz  1/1     Running   0          31s
stateless-app-79c79dd5-c6g2j  1/1     Running   0          31s
stateless-app-79c79dd5-hl67q  1/1     Running   0          20m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl autoscale deployment stateless-app --cpu-percent=50 --min=1 --max=5
horizontalpodautoscaler.autoscaling/stateless-app autoscaled
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa
NAME        REFERENCE   TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
stateless-app Deployment/stateless-app   cpu: 0%/50%   1       5       3       22s
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl describe hpa stateless-app
Name: stateless-app
Namespace: default
Labels: <none>
Annotations: <none>
CreationTimestamp: Mon, 27 Jan 2025 21:07:44 +0200
Reference: Deployment/stateless-app
Metrics: resource cpu on pods (as a percentage of request): 0% (0) / 50%
  Min replicas: 1
  Max replicas: 5
  Deployment pods: 3 current / 3 desired
Conditions:
  Type        Status  Reason           Message
  AbleToScale  True    ScaleDownStabilized recent recommendations were higher than current one, applying the highest recent recommendation
  ScalingActive True    ValidMetricFound   the HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of request)

```



load generation on pods

```
PS C:\Users\User> kubectl run -i --tty load-generator --image=busybox -- /bin/sh -c "while true; do wget -q -O- http://stateless-app.default.svc.cluster.local; done"
```

Autoscaler monitoring

```
PS C:\Users\User> kubectl get hpa -w
```

```
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family:Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">http://nginx.org/.<br/>
Commercial support is available at
<a href="http://nginx.com/">http://nginx.com/.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family:Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>

```

Practical Task 9: Rolling Update of an Application in AKS

Requirements:

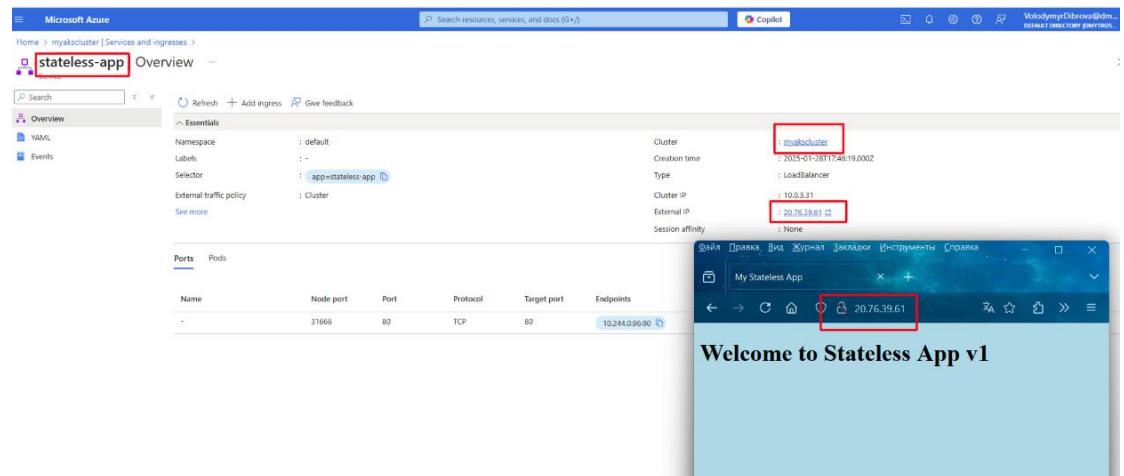
1. Deploy a lightweight version of your application to the AKS cluster.
 2. Update the Docker image to a new version with minimal changes (e.g., color change).
 3. Perform a rolling update using `kubectl set image` with minimal replicas to reduce resource usage.
 4. Verify the update process quickly and remove the deployment after the update.

1.Created a YAML file for deployment, service, and docker image with v1

The screenshot shows the Microsoft Azure Container Registry interface. At the top, there's a blue header bar with the Microsoft Azure logo and a search bar. Below it, the URL 'Home > myjavaapp' is visible. The main title is 'myjavaapp | Repositories'. On the left, a sidebar menu includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Quick start', 'Events', 'Settings', and 'Services'. Under 'Services', 'Repositories' is selected, highlighted with a red box. To the right, a list of repositories shows 'mywebdemo', 'node-web-app', and 'stateless-app', with 'stateless-app' also highlighted with a red box. Below the list is a code editor window displaying a simple HTML file:

```
<!DOCTYPE html>
<html>
<head>
    <title>My Stateless App</title>
    <style>
        body { background-color: lightblue; }
    </style>
</head>
<body>
    <h1>Welcome to Stateless App v1</h1>
</body>
</html>
```

2.Launched the service and opened external-IP in bowser



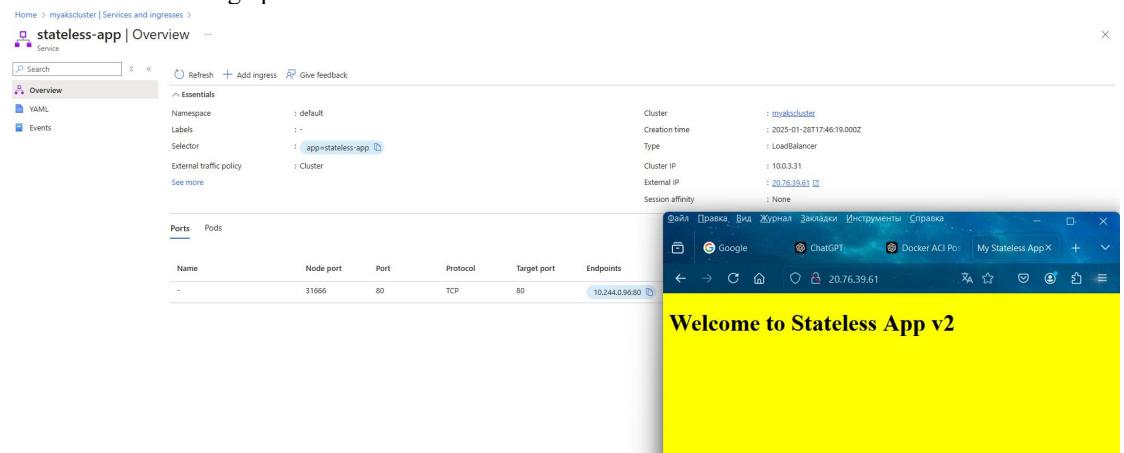
3. Created a new version of the image (v2) with minimal changes - updated body color to yellow

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>My Stateless App</title>
5      <style>
6          body { background-color: yellow; }
7      </style>
8  </head>
9  <body>
10     <h1>Welcome to Stateless App v2</h1>
11  </body>
12 </html>
13

```

4. Performed a rolling update



```

Windows PowerShell x + v
kubernetes   ClusterIP  10.0.3.1   <none>        443/TCP   57m
stateless-app LoadBalancer 10.0.3.31  28.76.39.61  80:31666/TCP 39m
PS E:\projects\terraform\pro\nodevwebapp\Rolling- kubectl describe deployment stateless-app
Name:           stateless-app
Namespace:      default
Labels:         app=stateless-app
Annotations:    deployment.kubernetes.io/revision: 4
                 kubernetes.io/change-cause: kubectl.exe set image deployment/stateless-app stateless-app=myjavaapp.azurecr.io/stateless-app:v2 --record=true
Selector:       app=stateless-app
Replicas:       1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=stateless-app
  Annotations: kubectl.kubernetes.io/restartedAt: 2025-01-28T20:06:11+02:00
  Containers:
    stateless-app:
      Image:  myjavaapp.azurecr.io/stateless-app:v2
      Port:   80/TCP
      Host Port:  8/TCP
      Limits:
        cpu:  500m
        memory:  256Mi
      Requests:
        cpu:  250m
        memory:  128Mi
      Environment:  <none>
      Mounts:  <none>
      Volumes:  <none>
      Node-selectors:  <none>
      Tolerations:  <none>
  Conditions:
    Type        Status  Reason
    Available  True    MinimumReplicasAvailable
    Progressing True    NewReplicaSetAvailable
OldReplicaSets: stateless-app-79c79dd5 (0/0 replicas created), stateless-app-5886cc7598 (0/0 replicas created), stateless-app-7f95cd4fb8 (0/0 replicas created)
NewReplicaSet:  stateless-app-85bb98cc85 (1/1 replicas created)
Events:
  Type        Reason     Age   From            Message
  Normal  ScalingReplicaSet 43m  deployment-controller  Scaled up replica set stateless-app-79c79dd5 to 1
  Normal  ScalingReplicaSet 24m  deployment-controller  Scaled up replica set stateless-app-5886cc7598 to 1
  Normal  ScalingReplicaSet 23m  deployment-controller  Scaled down replica set stateless-app-5886cc7598 to 0 from 1
  Normal  ScalingReplicaSet 23m  deployment-controller  Scaled up replica set stateless-app-7f95cd4fb8 to 1 from 0
  Normal  ScalingReplicaSet 16m  deployment-controller  Scaled down replica set stateless-app-79c79dd5 to 0 from 1
  Normal  ScalingReplicaSet 3m47s deployment-controller  Scaled up replica set stateless-app-85bb98cc85 to 1
  Normal  ScalingReplicaSet 3m43s deployment-controller  Scaled down replica set stateless-app-7f95cd4fb8 to 0 from 1

```

10:Bonus task. **GitOps with AKS Requirements:**

1. Setup ArgoCD on Azure Kubernetes Services
2. Perform image update on cluster

Actions Taken:

1. Created a namespace for ArgoCD and Installing ArgoCD in AKS

```

PS E:\projects\terraformProj\terraform-aks> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster
Merged "myakscluster" as current context in C:\Users\sergey\OneDrive\config
PS E:\projects\terraformProj\terraform-aks> kubectl get nodes
NAME           STATUS   ROLES    AGE     VERSION
aks-default-283137413-vwss000000 Ready    <none>   16h    v1.30.7
aks-default-283137413-vwss000001 Ready    <none>   16h    v1.30.7
PS E:\projects\terraformProj\terraform-aks> kubectl create namespace argocd
namespace/argocd created
PS E:\projects\terraformProj\terraform-aks> kubectl apply -n argocd --f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/appprojects.argoproj.io created
serviceaccount/argocd-application-controller created
serviceaccount/argocd-applicationset-controller created
serviceaccount/argocd-dex-server created
serviceaccount/argocd-notifications-controller created
serviceaccount/argocd-redis created
serviceaccount/argocd-repo-server created
serviceaccount/argocd-server created
role.rbac.authorization.k8s.io/argocd-application-controller created
role.rbac.authorization.k8s.io/argocd-applicationset-controller created
role.rbac.authorization.k8s.io/argocd-dex-server created
role.rbac.authorization.k8s.io/argocd-notifications-controller created
role.rbac.authorization.k8s.io/argocd-redis created
role.rbac.authorization.k8s.io/argocd-server created
clusterrole.rbac.authorization.k8s.io/argocd-application-controller created
clusterrole.rbac.authorization.k8s.io/argocd-applicationset-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-server created
clusterrolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-server created
configmap/argocd-cm created
configmap/argocd-cmd-params-cm created
configmap/argocd-gpg-keys-cm created
configmap/argocd-notifications-cm created

```

2. Temporary access via port-forward

```

Error from server (BadRequest): error creating object: services "argo-cd-server" is invalid: services.spec.ports[0].port: Invalid value: "8080": must be in range [1024, 65535]
Error from server (NotFound): services "argo-cd-server" not found
PS E:\projects\terraformProj\terraform-aks> kubectl delete service argocd-server
Error from server (BadRequest): error deleting object: services "argo-cd-server" deleted
PS E:\projects\terraformProj\terraform-aks> kubectl expose deployment argocd-server --name=argocd --type=LoadBalancer --port=443 --target-port=8080
service/argocd-server exposed
PS E:\projects\terraformProj\terraform-aks> kubectl get svc -n argocd
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
argo-cd-server   LoadBalancer   10.0.7.158   <none>       443:31777/TCP   7s
PS E:\projects\terraformProj\terraform-aks> kubectl get svc -n argocd
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
argo-cd-server   LoadBalancer   10.0.7.158   <none>       443:31777/TCP   7s
PS E:\projects\terraformProj\terraform-aks> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from [::]:8080 -> 8080
Forwarding from [::]:8080 -> 8080
Handling connection for 8080
E0129 12:56:51.932264 15232 portforward.go:385] error copying from remote stream to local connection: readfrom tcp4 127.0.0.1:8080->127.0.0.1:64770: write
tcp4 127.0.0.1:8080->127.0.0.1:64770: wsasend: An established connection was aborted by the software in your host machine.
PS E:\projects\terraformProj\terraform-aks> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from [::]:8080 -> 8080
Forwarding from [::]:8080 -> 8080
Handling connection for 8080
Handling connection for 8080

```

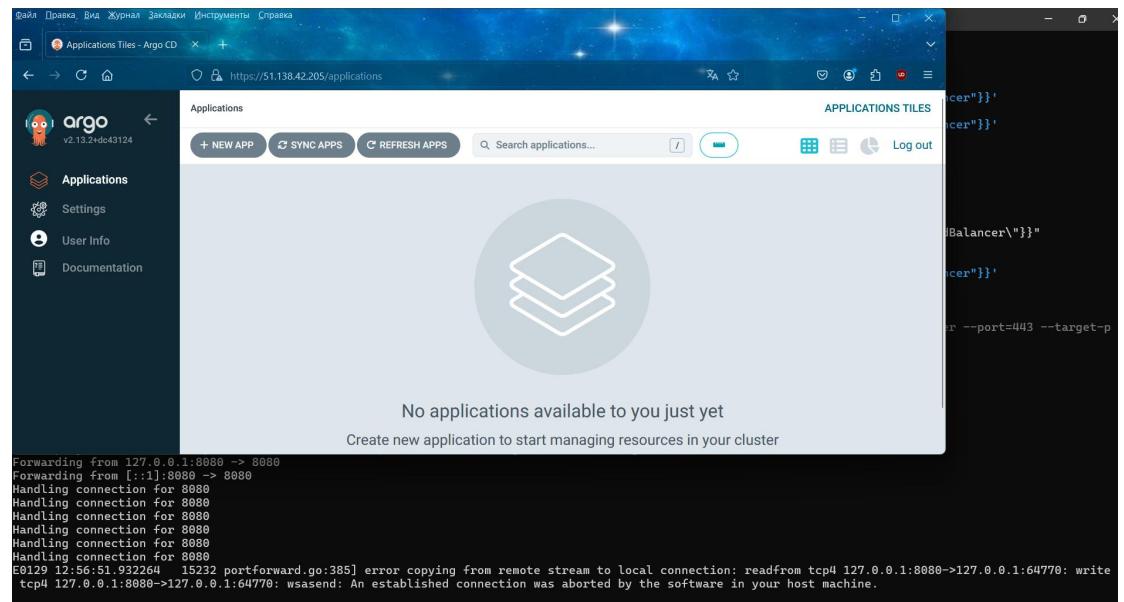
3. Deleted the old service and created new service LoadBalancer

```

PS E:\projects\terraformProj\terraform-aks> kubectl delete svc argocd-server -n argocd
service "argocd-server" deleted
PS E:\projects\terraformProj\terraform-aks> kubectl expose deployment argocd-server --name=argocd --type=LoadBalancer --port=443 --target-port=8080
service/argocd-server exposed
PS E:\projects\terraformProj\terraform-aks> kubectl get svc -n argocd argocd-server
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
argo-cd-server   LoadBalancer   10.0.7.158   <none>       443:31777/TCP   7s
PS E:\projects\terraformProj\terraform-aks> kubectl get svc -n argocd argocd-server
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
argo-cd-server   LoadBalancer   10.0.7.158   51.138.42.205   443:31777/TCP   14s
PS E:\projects\terraformProj\terraform-aks> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::]:8080 -> 8080
Handling connection for 8080
E0129 12:56:51.932264 15232 portforward.go:385] error copying from remote stream to local connection: readfrom tcp4 127.0.0.1:8080->127.0.0.1:64770: write
tcp4 127.0.0.1:8080->127.0.0.1:64770: wsasend: An established connection was aborted by the software in your host machine.

```

4. Open with external ip:



5. connect CLI to ArgoCD server

```

PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd version
argocd: v2.13.3+a25c8a0
  BuildDate: 2025-01-03T19:06:52Z
  GitCommit: a25c8a0eef7830be0c2c9074c92dbea8ff23a962
  GitTreeState: clean
  GoVersion: go1.22.10
  Compiler: gc
  Platform: windows/amd64
time="2025-01-29T13:24:46+02:00" level=fatal msg="Argo CD server address unspecified"
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd login 51.138.42.205 --username admin --password -Qa3rijL0fPrKbRH --insecure
'admin:login' logged in successfully
Context '51.138.42.205' updated
PS E:\projects\terraformProj\nodeWebApp\Rolling> |

```

Name	Node port	Port	Protocol	Target port	Endpoints
-	31772	443	TCP	8080	10.244.0.156:8080

6. synch in gitops

```

PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd app create my-app ` 
>>   --repo https://github.com/Vivien87/mywebdemo.git ` 
>>   --path . ` 
>>   --dest-server https://kubernetes.default.svc ` 
>>   --dest-namespace default ` 
application 'my-app' created
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd app get my-app
Name:                  argocd/my-app
Project:               default
Server:                https://kubernetes.default.svc
Namespace:             default
URL:                  https://51.138.42.205/applications/my-app
Source:
- Repo:                https://github.com/Vivien87/mywebdemo.git
  Target:              .
  Path:                .
SyncWindow:            Sync Allowed
Sync Policy:           Manual
Sync Status:           Synced to  (9d82e02)
Health Status:         Healthy
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd app sync my-app

Name:                  argocd/my-app
Project:               default
Server:                https://kubernetes.default.svc
Namespace:             default
URL:                  https://51.138.42.205/applications/my-app
Source:
- Repo:                https://github.com/Vivien87/mywebdemo.git
  Target:              .
  Path:                .
SyncWindow:            Sync Allowed
Sync Policy:           Manual
Sync Status:           Synced to  (9d82e02)
Health Status:         Healthy

Operation:             Sync
Sync Revision:         9d82e02e03a4c6bf26b2f763d11ee3bd4dba1979
Phase:                 Succeeded
Start:                 2025-01-29 13:37:11 +0200 EET
Finished:              2025-01-29 13:37:11 +0200 EET
Duration:              0s
Message:               successfully synced (no more tasks)
PS E:\projects\terraformProj\nodeWebApp\Rolling> |

```

The screenshot shows the Argo CD web interface with the following details:

- Project:** default
- Labels:** healthy, Synced
- Repository:** https://github.com/Vivien87/mywebdem...
- Target Ref:** HEAD
- Path:** .
- Destination:** in-cluster
- Namespace:** default
- Created:** 01/29/2025 13:36:14 (9 minutes ago)
- Last Sync:** 01/29/2025 13:37:11 (8 minutes ago)

Buttons at the bottom: SYNC, REFRESH, DELETE.

The screenshot shows the Microsoft Azure Kubernetes Services Overview page for a secret named 'repo-1295282847'. The URL field contains a long Base64 encoded string: 'aHR0cHM6Ly9naXRodWIuY29tL1Zpdmllbjg3L215d2VzGvtby5naXQ='.

```
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd repo list
TYPE NAME REPO INSECURE OCI LFS CREDs STATUS MESSAGE PROJECT
git https://github.com/Vivien87/symwebdemo.git false false false false Successful
https://github.com/Vivien87/symwebdemo.git [System.Text.Encoding]:UTF8.GetString([System.Convert]::FromBase64String("aHR0cHM6Ly9naXRodWIuY29tL1Zpdmllbjg3L215d2VzGvtby5naXQ="))
PS E:\projects\terraformProj\nodeWebApp\Rolling>
```